

# The $q$ -gradient method for global optimization

Aline C. Soterroni · Roberto L. Galski ·  
Fernando M. Ramos

Received: date / Accepted: date

**Abstract** The  $q$ -gradient is an extension of the classical gradient vector based on the concept of Jackson's derivative. Here we introduce the  $q$ -gradient method for unconstrained global optimization. The main idea behind the method is the use of the negative of the  $q$ -gradient of the objective function as the search direction. In this sense, the method here proposed is a generalization of the well-known steepest descent method. The use of Jackson's derivative has shown to be an effective mechanism for escaping from local minima. The algorithm for the  $q$ -gradient method is complemented with strategies to generate the parameter  $q$  and to compute the step length in a way that the search process gradually shifts from global in the beginning to almost local search in the end. For testing this new approach, we considered six commonly used test functions and compared our results with a deterministic global optimization solver and three Genetic Algorithms (GAs) considered effective in optimizing multidimensional unimodal and multimodal functions. For the multimodal test functions, the  $q$ -gradient method outperformed the solver and the GAs, reaching the minimum, respectively, with a better accuracy and with less function evaluations.

**Keywords** steepest descent method · Jackson's derivative ·  $q$ -gradient ·  $q$ -gradient method

## 1 Introduction

Over the last decades the  $q$ -calculus has been connecting mathematics and physics in applications that span from quantum theory and statistical me-

---

Aline C. Soterroni · Roberto L. Galski · Fernando M. Ramos  
Applied Computing Program · Satellite Control Centre · Laboratory of Computing and Applied Mathematics  
National Institute for Space Research, São José dos Campos, 12227-010, Brazil  
E-mail: alinecsoterroni@gmailcom, galski@ccs.inpe.br, fernando.ramos@inpe.br

chanics, to number theory and combinatorics (see [1] and references therein). Its history dates back to the beginnings of the last century when, based on pioneering works of Euler and Heine, the English reverend Frank Hilton Jackson developed the  $q$ -calculus in a systematic way [2]. His work gave rise to generalizations of series, functions and special numbers within the context of the  $q$ -calculus [3]. More important, he introduced the concepts of the  $q$ -derivative [4] (also known as Jackson's derivative) and the  $q$ -integral [5].

The  $q$ -derivative of a function  $f$  is defined as [6]

$$D_q f(x) = \begin{cases} \frac{f(qx) - f(x)}{qx - x}, & x \neq 0 \\ \frac{df(0)}{dx}, & x = 0 \end{cases} \quad (1)$$

where  $q$  is a real number different from 1. In the limiting case of  $q \rightarrow 1$ , the  $q$ -derivative reduces to the classical derivative.

Let  $f(x) = x^n$ , for example. In this case, the classical derivative of  $f$  is  $nx^{n-1}$  and the  $q$ -derivative is  $[n]x^{n-1}$ , where  $[n]$  is the  $q$ -analogue of  $n$  given by

$$[n] = \frac{q^n - 1}{q - 1}.$$

As  $q \rightarrow 1$ ,  $[n]$  tends to  $n$ . This definition is used to calculate the  $q$ -binomial and establish a  $q$ -analogue of Taylor's formula that encompasses many results such as the Euler's identities for  $q$ -exponential functions, Gauss's  $q$ -binomial formula, Heine's formula for a  $q$ -hypergeometric function, among others mathematical results [7].

Considering arbitrary functions  $f(x)$  and  $g(x)$ , the  $q$ -derivative operator satisfy the following properties [7]:

- 1) The  $q$ -derivative is a linear operator for any constants  $a$  and  $b$

$$D_q(af(x) + bg(x)) = aD_q f(x) + bD_q g(x).$$

- 2) The  $q$ -derivative of the product of  $f(x)$  and  $g(x)$  is given by

$$D_q(f(x)g(x)) = f(qx)D_q g(x) + g(x)D_q f(x)$$

that, by symmetry, is equivalent to

$$D_q(f(x)g(x)) = f(x)D_q g(x) + g(qx)D_q f(x).$$

- 3) The  $q$ -derivative of the quotient of  $f(x)$  and  $g(x)$  is calculated as

$$D_q \left( \frac{f(x)}{g(x)} \right) = \frac{g(x)D_q f(x) - f(x)D_q g(x)}{g(x)g(qx)}$$

or equivalently

$$D_q \left( \frac{f(x)}{g(x)} \right) = \frac{g(qx)D_q f(x) - f(qx)D_q g(x)}{g(x)g(qx)}.$$

The chain rule for  $q$ -derivatives does not exist, except for a function of the form  $f(u(x))$ , where  $u(x) = \alpha x^\beta$ , with  $\alpha, \beta$  being constants. More details on the properties of  $q$ -derivatives can be found in [7].

Let a general nonlinear unconstrained optimization problem be defined as

$$\min F(\mathbf{x}), \quad \mathbf{x} = (x_1, \dots, x_i, \dots, x_n) \quad (2)$$

where  $\mathbf{x} \in \mathbb{R}^n$  is the vector of the independent variables and  $F : \mathbb{R}^n \rightarrow \mathbb{R}$  is the objective function. The steepest descent method (also known as the gradient descent method) uses information on the gradient of the objective function in seeking the optimum. The search direction is given by the negative of the gradient of  $F$ . This search strategy is an obvious choice since along this direction is where the objective function decreases most rapidly.

Requiring only information about first-derivatives, the steepest descent method is attractive because of its limited computational cost and storage requirements [8]. However, for multimodal functions, unless one knows in advance where to start from, the search procedure frequently gets stuck in one of the local minima. Consequently, the steepest descent method is not recommended for real-world optimization problems that are usually multimodal. Nevertheless, because of its inherent simplicity, it represents a good starting point for the development of more advanced optimization methods.

Here we propose a generalization of the steepest descent method in which the gradient of the objective function is replaced by its  $q$ -analogue. Accordingly, the search direction is taken as the negative of the  $q$ -gradient of  $F$ . For  $q = 1$ , the here called  $q$ -gradient method reduces to the classical steepest descent method. In order to evaluate the performance of the  $q$ -gradient method we consider three unimodal and three multimodal test functions commonly used as benchmarks. We compare our results with those obtained with the Genetic Algorithms (GAs) G3-PCX developed by Deb et al. [9], and the SPC-vSBX and SPC-PNX developed by Ballester and Carter [10], which previous studies have shown to be effective in minimizing multidimensional unimodal and multimodal function. The best values generated by our approach are also compared with those given by the well-known LINDOGlobal deterministic solver available in NEOS Server for Optimization [11].

The rest of the paper is organized as follows. In Section 2 the  $q$ -gradient vector is defined and its properties are presented. In Section 3 the strategies to obtain the parameter  $q$  and the step length are described, and an algorithm for the  $q$ -gradient method is proposed. Section 4 describes the computational experiments and Section 5 discusses the results. Finally, in Section 6 some conclusions and future work are presented.

## 2 The $q$ -Gradient

Given a differentiable function of  $n$  variables  $F(\mathbf{x})$ , the gradient of  $F$  is the vector of the  $n$  first-order partial derivatives of  $F$ . Similarly, the  $q$ -gradient is the vector of the  $n$  first-order partial  $q$ -derivatives of  $F$ . Thus, let the parameter

$q$  be a vector  $\mathbf{q} = (q_1, \dots, q_i, \dots, q_n)$ , where  $q_i \neq 1 \ \forall i$ , the first-order partial  $q$ -derivative with respect to the variable  $x_i$  is given by

$$D_{q_i, x_i} F(\mathbf{x}) = \frac{F(x_1, \dots, q_i x_i, \dots, x_n) - F(x_1, \dots, x_i, \dots, x_n)}{q_i x_i - x_i} \quad (3)$$

with

$$D_{q_i, x_i} F(\mathbf{x})|_{x_i=0} = \frac{\partial F(\mathbf{x})}{\partial x_i} \quad (4)$$

and

$$D_{q_i, x_i} F(\mathbf{x})|_{q_i=1} = \frac{\partial F(\mathbf{x})}{\partial x_i}. \quad (5)$$

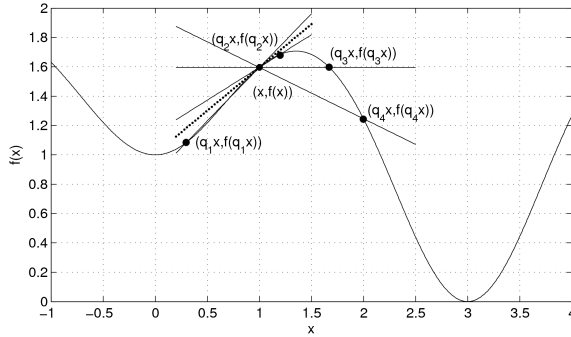
This framework can be extended to define the  $q$ -gradient of a function of  $n$  variables as

$$\nabla_{\mathbf{q}} F(\mathbf{x}) = [D_{q_1, x_1} F(\mathbf{x}) \ \dots \ D_{q_i, x_i} F(\mathbf{x}) \ \dots \ D_{q_n, x_n} F(\mathbf{x})] \quad (6)$$

with the classical gradient being recovered in the limit of  $q_i \rightarrow 1$ , for all  $i = 1, \dots, n$ .

Let  $f(x)$  be a function of one variable. In this case, the geometric interpretation of the gradient is simply the slope of the tangent line at  $x$ . Similarly, the  $q$ -gradient of  $f$  has also a straightforward geometric interpretation as the slope of the secant line passing through the points  $(x, f(x))$  and  $(qx, f(qx))$ . If the slope of the secant line is positive (negative), the  $q$ -gradient points to the right (left) direction. Fig. 1 illustrates this geometric interpretation for

$$f(x) = 2 - (e^{-x^2} + 2e^{-(x-3)^2}). \quad (7)$$



**Fig. 1** Geometric interpretation of the classical derivative (dotted line) and the  $q$ -derivative of  $f(x)$  at  $x = 1.0$  and different values of the parameter  $q$ .

Since the slope of the tangent line (dotted line) at  $x = 1$  is positive, the steepest descent method at this point will move necessarily to the left and, thus, will be trapped by the local minimum at  $x = 0$ . The slope of the secant

line passing through the points  $(x, f(x))$  and  $(qx, f(qx))$  can be positive or negative depending on the value of the parameter  $q$ . For instance, if  $q = 2$  the  $q$ -derivative is negative at  $x = 1$  (see the secant line passing through  $(x, f(x))$  and  $(q_4x, f(q_4x))$  in Fig. 1), which potentially allows a minimization strategy based on the value of the  $q$ -gradient to take a leap to the right, towards the global minimum of  $f$ . Note that there is a value  $1.5 < q_3 < 2$  for which  $x = 1$  is a stationary point of the  $q$ -gradient ( $\nabla_q F(x) = 0$ ) but that does not correspond to any minimum or maximum of  $f$ . Finally, for  $0 < q_1 < 0.5$  or  $1 < q_2 < 1.5$  the slope of the secant line is positive and the  $q$ -gradient method will move to the left as the steepest descent method.

The simple example above shows that the use of the  $q$ -gradient offers a new mechanism to escape from local minima. Moreover, the transition from global to local search might be controlled by the parameter  $q$ , provided a suitable strategy for generating  $q$ -values is incorporated into the minimization algorithm.

### 3 $q$ -Gradient Method Description

A general optimization strategy is to consider an iterative procedure that, starting from  $\mathbf{x}^0$ , generates a sequence  $\{\mathbf{x}^k\}$  given by [12]

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{d}^k \quad (8)$$

where  $k$  is the iteration number,  $\mathbf{d}^k$  is the search direction and  $\alpha^k$  is the step length or the distance moved along  $\mathbf{d}^k$  in the iteration  $k$ .

Optimization methods can be characterized according to the direction and step length used in (8). The steepest descent method sets  $\mathbf{d}^k = -\nabla F(\mathbf{x}^k)$  and the step length  $\alpha^k$  is usually determined by a line-search technique that minimizes the objective function along the direction  $\mathbf{d}^k$ . In the  $q$ -gradient method, as here proposed, the search direction is the negative of the  $q$ -gradient of the objective function  $-\nabla_q F(\mathbf{x})$ . Thus the optimization procedure defined by (8) becomes

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha^k \nabla_q F(\mathbf{x}^k). \quad (9)$$

Key to the performance of the  $q$ -gradient method, the strategies used to specify the parameter  $q$  and the step length  $\alpha$  are described below.

#### 3.1 Parameter $\mathbf{q}$

Considering a function of  $n$  variables  $F(\mathbf{x})$ , a set of  $n$  different parameters  $q_i \in \mathbb{R} - \{1\}$  ( $i = 1, \dots, n$ ) are needed to compute the  $q$ -gradient vector of  $F$ . The overall strategy adopted here is to draw the values of  $q_i$  (or some variable related to them) from some suitable probability density function (pdf), and with a standard deviation that decreases as the iterative search proceeds. In this sense, the role of the standard deviation here is reminiscent of the one played by the temperature in a simulated annealing (SA) algorithm, that is,

to make the algorithm go from a very random (at the beginning) to a very deterministic search (at the end).

In the current implementation we opted to first draw the values of  $q_i^k x_i^k$  from a Gaussian pdf given by

$$f(x) = \frac{1}{\sigma\sqrt{(2\pi)}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right], \quad (10)$$

with  $\mu = x_i^k$  and  $\sigma = \sigma^k$ ; then, we computed the values of  $q_i^k$ .

Starting from  $\sigma^0$ , the standard deviation of the pdf is decreased by the following “cooling” schedule,  $\sigma^{k+1} = \beta \cdot \sigma^k$ , where  $0 < \beta < 1$  is the reduction factor. As  $\sigma^k$  approaches zero, the values of  $q_i^k$  tend to unity, the algorithm reduces to the steepest descent method, and the search process becomes essentially local. As in a SA algorithm, the performance of the minimization algorithm depends crucially on the choice of parameters  $\sigma^0$  and  $\beta$ . A too rapid decrease of  $\sigma^k$ , for example, may cause the algorithm to be trapped in a local minimum.

### 3.2 Step Length

The calculation of the step length  $\alpha$  is a tradeoff. On the one hand,  $\alpha$  should give a considerable reduction of the objective function. On the other hand, its calculation should not take too many evaluations of  $F$  [13]. Steepest descent algorithms generally use line-search techniques to determine the step length  $\alpha^k$  along the steepest descent direction  $\mathbf{d}^k = -\nabla F(\mathbf{x}^k)$  at the iteration  $k$ . A first version of our algorithm [14] applied the golden section for step length determination. However, traditional line-search algorithms, like the golden section, ensure that the condition  $F(\mathbf{x}^{k+1}) < F(\mathbf{x}^k)$  is always satisfied, what obviously is a poor strategy when dealing with multimodal minimization problems. In addition, depending on the value of  $q$ , the negative of the  $q$ -gradient may not point to the local descent direction.

One way to circumvent these problems is to use a diminishing step length  $\alpha^k$ , i.e., the initial step length  $\alpha^0$  is reduced by  $\alpha^{k+1} = \beta \cdot \alpha^k$ , where, for the sake of simplicity,  $\beta$  is the same reduction factor used to compute  $\sigma^k$ . As the step length decreases (and the values of  $q_i^k$ , in parallel, tend to unity), a smooth transition to an increasingly local search process occurs.

### 3.3 Algorithm for the $q$ -Gradient Method

Based on the definitions presented in the previous sections, the  $q$ -gradient method for continuous global optimization problems is described as follows.

#### ALGORITHM 1 ( $q$ -Gradient Method)

Given  $\mathbf{x}^0$  (initial point),  $\sigma^0 > 0$ ,  $\alpha^0 > 0$  and  $0 < \beta < 1$ :

- 1) Set  $k = 0$
- 2) Set  $\mathbf{x}_{best} = \mathbf{x}^k$
- 3) While the stopping criteria are not reached, do:
  - a) Generate  $\mathbf{q}^k \mathbf{x}^k$  by a Gaussian distribution with  $\sigma^k$  and  $\mu^k = \mathbf{x}^k$
  - b) Calculate the  $q$ -gradient  $\nabla_q F(\mathbf{x}^k)$  by (6)
  - c) Set  $\mathbf{d}^k = -\nabla_q F(\mathbf{x}^k) / \|\nabla_q F(\mathbf{x}^k)\|$
  - d) Set  $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \cdot \mathbf{d}^k$
  - e) If  $F(\mathbf{x}^{k+1}) < F(\mathbf{x}_{best})$  set  $\mathbf{x}_{best} = \mathbf{x}^{k+1}$
  - f) Set  $\sigma^{k+1} = \beta \cdot \sigma^k$  and  $\alpha^{k+1} = \beta \cdot \alpha^k$
  - g) Set  $k = k + 1$
- 4) Return  $\mathbf{x}_{best}$  and  $F(\mathbf{x}_{best})$ .

The main idea behind the  $q$ -gradient method is to use the negative of the  $q$ -gradient of  $F$ , instead of the negative of the classical gradient of  $F$ , as the search direction. Strategies for generating the vector  $\mathbf{q}^k$  and the step length  $\alpha^k$ , at each iteration, complement this very simple algorithm.

Basically, there are three free parameters to be specified, namely,  $\sigma^0$ ,  $\alpha^0$  and  $\beta$ . The initial standard deviation  $\sigma^0$  determines how stochastic is the search. For multimodal functions, it must be high enough to allow the method to properly sample the search space. The reduction factor  $\beta$  controls the speed of the transition from stochastic to deterministic search. A  $\beta$  close to 1 reduces the risk of being trapped in a local minimum. The last free parameter, the initial step length  $\alpha^0$ , depends heavily on the topology of the search space and, thus, requires some empirical exploration. In the end, as with the choice of the cooling schedule in a SA algorithm [15], an appropriate specification of the three free parameters is strictly dependent on the objective function. Although a bad choice may lead to some deterioration in its performance, the  $q$ -gradient method has shown to be sufficiently robust to still be capable of reaching the global minimum.

The algorithm stops when the appropriate stopping criterium is attained. In real-world applications (i.e., in problems for which the global minimum is not known), it can be the maximum number of function evaluations, or the value of the local gradient  $\|\nabla F(\mathbf{x}^k)\| < \epsilon$ , since the  $q$ -gradient method converges to the steepest descent method at the end of the search. The algorithm returns the  $\mathbf{x}_{best}$  as the minimum value of the objective function  $F$  obtained during the iterative procedure, i.e.,  $F(\mathbf{x}_{best}) \leq F(\mathbf{x}^k)$ ,  $\forall k$ .

## 4 Computational Experiments

The performance of the  $q$ -gradient method was evaluated over six 20-variable test functions ( $n = 20$ ) commonly employed in the literature. We use the same experimental setup and stopping criteria as described in [9] and [10] in order to allow a direct comparison with their results. The stopping criteria are: maximum of  $10^6$  function evaluations or  $F(\mathbf{x}) < 10^{-20}$ .

As in [9] and [10], we set the three free parameters in the  $q$ -gradient algorithm after preliminary exploratory runs. The results presented here are for those which yielded the best performance. The benchmark consists of the following analytical functions:

- 1) Ellipsoidal function ( $F_{elp}$ )

$$F_{elp} = \sum_{i=1}^n i \mathbf{x}_i^2. \quad (11)$$

Although the Ellipsoidal function is convex and unimodal, it is an example of a poorly scaled function.

- 2) Schwefel's function ( $F_{sch}$ )

$$F_{sch} = \sum_{i=1}^n \left( \sum_{j=1}^i \mathbf{x}_j \right)^2. \quad (12)$$

The Schwefel's function is an extension of the Ellipsoidal function and it is also a unimodal and poorly scaled function.

- 3) Generalized Rosenbrock's function ( $F_{ros}$ )

$$F_{ros} = \sum_{i=1}^{n-1} [100 \cdot (\mathbf{x}_i^2 - \mathbf{x}_{i+1})^2 + (1 - \mathbf{x}_i)^2]. \quad (13)$$

Although the Rosenbrock's function is a well-known unimodal function for  $n = 2$ , numerical experiments have shown that for  $4 \leq n \leq 30$  the function has two minima, the global one at  $\mathbf{x} = \mathbf{1}$  and a local minimum that changes with the dimensionality  $n$  [16]. The Rosenbrock's function is considered a test case for premature convergence once the global minimum lays inside a long, narrow, and parabolic shaped flat valley.

- 4) Ackley's function ( $F_{ackl}$ )

$$F_{ackl} = 20 + e - 20 \exp \left( -0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n \mathbf{x}_i^2} \right) - \exp \left( \frac{1}{n} \sum_{i=1}^n \cos(2\pi \mathbf{x}_i) \right). \quad (14)$$

The Ackley's function is highly multimodal and the basin of the local minima increase in size as one moves away from the global minimum [10].

- 5) Rastrigin's function ( $F_{rtg}$ )

$$F_{rtg} = 10n + \sum_{i=1}^n (\mathbf{x}_i^2 - 10 \cos(2\pi \mathbf{x}_i)). \quad (15)$$



The Rastrigin's function has a parabolic landscape away from the global minimum, but as we move towards the global minimum, the size of the basins increase. The function is highly multimodal and its characteristics are known to be difficult for many optimization algorithms to achieve the global minimum [10].

6) Rotated Rastrigin's function ( $F_{rrtg}$ ).

$$F_{rrtg} = 10n + \sum_{i=1}^n (\mathbf{y}_i^2 - 10 \cos(2\pi \mathbf{y}_i)), \quad \mathbf{y} = A \cdot \mathbf{x} \quad (16)$$

$$A_{i,i} = 4/5$$

$$A_{i,i+1} = 3/5 \text{ (} i \text{ odd)}$$

$$A_{i,i-1} = -3/5 \text{ (} i \text{ even)}$$

$$A_{i,j} = 0 \text{ (otherwise)}$$

The rotated Rastrigin's is a highly multimodal function without local minima arranged along the axis [10].

For all these functions the global minimum is  $F(\mathbf{x}^*) = 0$  at  $\mathbf{x}^* = \mathbf{0}$ , except the Generalized Rosenbrock's function where  $\mathbf{x}^* = \mathbf{1}$ . The initial point set  $\mathbf{x}^0$  for each function is generated by a uniform distribution within  $[-10, -5]$ , as used in [9] and [10].

## 5 Results

### 5.1 Comparison with GAs

Extensive comparisons between the GAs G3-PCX ([9] for Ellipsoidal, Schwefel, Rosenbrock and Rastrigin functions; [10] for Ackley and Rotated Rastrigin), SPC-vSBX and SPC-PNX [10] and the  $q$ -gradient method are presented in Tables 2 and 3. As in [9] and [10], the best, median and worst columns refer to the number of function evaluations required to reach the accuracy  $10^{-20}$ . When this condition is not achieved, the best value found so far for the test function after  $10^6$  evaluations is given. The column "Success" refers to how many runs reached the target accuracy, for unimodal functions, or ended up within the global minimum basin, for multimodal ones. The best performances are highlighted in bold in each table. The corresponding values of the best parameters  $\sigma^0$ ,  $\alpha^0$  and  $\beta$  used in each test function are given in Table 1.

In Table 2, for the Ellipsoidal function, the  $q$ -gradient method achieved the required accuracy  $10^{-20}$  for all 50 runs, with an overall performance similar to the one displayed by the G3-PCX, the best algorithm among the GAs. As for the Schwefel's function, the  $q$ -gradient method again attained the required accuracy for all runs but was outperformed by the G3-PCX in terms of the number of function evaluations. Finally, for the Rosenbrock's function,

**Table 1** Parameters used by the  $q$ -gradient method over the test functions.

Functions	$\sigma^0$	$\alpha^0$	$\beta$
Ellipsoidal	0.4	38	0.86
Schwefel	0.1	1	0.997
Rosenbrock	0.1	0.1	0.9995
Ackley	20	12	0.90
Rastrigin	21	0.3	0.9995
Rotated Rastrigin	30	0.5	0.999

**Table 2** Performance comparison between G3-PCX, SPC-vSBX, SPC-PNX and  $q$ -gradient method over the unimodal test functions in terms of the best, median and worst number of function evaluations required to reach the accuracy  $10^{-20}$ . The column “Success” refers to how many runs reached the target accuracy.

Function	Method	Best	Median	Worst	$\mathbf{x}_{best}$	Success
Ellipsoidal	<b>G3-PCX</b>	<b>5,826</b>	<b>6,800</b>	<b>7,728</b>	<b><math>10^{-20}</math></b>	<b>10/10</b>
	SPC-vSBX	49,084	50,952	57,479	$10^{-20}$	10/10
	SPC-PNX	36,360	39,360	40,905	$10^{-20}$	10/10
	<b><math>q</math>-Gradient</b>	<b>5,905</b>	<b>7,053</b>	<b>7,381</b>	<b><math>10^{-20}</math></b>	<b>50/50</b>
Schwefel	<b>G3-PCX</b>	<b>13,988</b>	<b>15,602</b>	<b>17,188</b>	<b><math>10^{-20}</math></b>	<b>10/10</b>
	SPC-vSBX	260,442	294,231	334,743	$10^{-20}$	10/10
	SPC-PNX	236,342	283,321	299,301	$10^{-20}$	10/10
	$q$ -Gradient	289,174	296,103	299,178	$10^{-20}$	50/50
Rosenbrock	<b>G3-PCX</b>	<b>16,508</b>	<b>21,452</b>	<b>25,520</b>	<b><math>10^{-20}</math></b>	<b>36/50</b>
	SPC-vSBX	$10^6$	-	-	$10^{-4}$	48/50
	SPC-PNX	$10^6$	-	-	$10^{-10}$	38/50
	$q$ -Gradient	$10^6$	-	-	$10^{-10}$	50/50

the  $q$ -gradient was beaten by the G3-PCX (the only to achieve the required accuracy) but performed better then the two other GAs.

The overall evaluation of the  $q$ -gradient method performance in these numerical experiments with unimodal (or quasi-unimodal) test functions indicates that it reaches the required accuracy (or the minimum global basin) in 100% of the runs, but it is not faster than the G3-PCX. This picture improves a lot when it comes to tackle the multimodal Ackley’s and Rastrigin’s functions.

In Table 3, due to limited computing precision the required accuracy for the Ackley’s function was set to  $10^{-10}$  for the GAs and  $10^{-15}$  in our simulations<sup>1</sup>. The  $q$ -gradient method was here clearly better than the GAs, reaching the required accuracy in more runs or in less functions evaluations. For the

<sup>1</sup> Numerical experiments have shown that Ackley’s function evaluated at  $\mathbf{x} = 0$  with double precision is equal to  $-0.4440892098500626\text{E}-015$  and not zero.

**Table 3** Performance comparison between G3-PCX, SPC-vSBX, SPC-PNX and  $q$ -gradient method over the multimodal test functions in terms of the best, median and worst number of function evaluations required to reach the accuracy  $10^{-20}$ . The column “Success” refers to how many runs reached the target accuracy.

Function	Method	Best	Median	Worst	$\mathbf{x}_{best}$	Success
Ackley	G3-PCX	$10^6$	-	-	3,959	0
	SPC-vSBX	57,463	63,899	65,902	$10^{-10}$	10/10
	SPC-PNX	45,736	48,095	49,392	$10^{-10}$	10/10
	<b><math>q</math>-Gradient</b>	<b>11,850</b>	<b>12,465</b>	<b>13,039</b>	<b><math>10^{-15}</math></b>	<b>50/50</b>
Rastrigin	G3-PCX	$10^6$	-	-	15,936	0
	SPC-vSBX	260,685	306,819	418,482	$10^{-20}$	6/10
	SPC-PNX	$10^6$	-	-	4.975	0
	<b><math>q</math>-Gradient</b>	<b>676,050</b>	<b>692,450</b>	<b>705,037</b>	<b><math>10^{-20}</math></b>	<b>48/50</b>
Rotated Rastrigin	G3-PCX	$10^6$	-	-	309.429	0
	SPC-vSBX	$10^6$	-	-	8.955	0
	SPC-PNX	$10^6$	-	-	3.980	0
	<b><math>q</math>-Gradient</b>	<b>541,857</b>	<b>545,957</b>	<b>549,114</b>	<b><math>10^{-20}</math></b>	<b>20/50</b>

Rastrigin’s function, the G3-PCX and the SPC-PNX were unable to attain the global minimum basin. The other two algorithms reached the required accuracy  $10^{-20}$ , but the  $q$ -gradient method was the only to do it in 96% of the runs (48 over 50). Finally, in the case of the rotated Rastrigin’s function, the  $q$ -gradient was the only algorithm to reach the minimum, attaining the required accuracy in 20 out of 50 independent runs. Summarizing the results with multimodal functions, we may say that the  $q$ -gradient method outperformed the GAs in all the three test cases considered, reaching the minimum with less function evaluations or in more independent runs.

## 5.2 Comparison with Deterministic Solver

In order to compare our approach with deterministic global optimization, the test functions were implemented in GAMS format [17] and solved using the LINDOGlobal solver through the version freely available on NEOS Server for Optimization [11]. The optimization procedure of the LINDOGlobal solver employs branch-and-cut methods to break an nonlinear model down into a list of subproblems. In addition, the solver has a multistart feature that restarts the standard (non-global) nonlinear solver from a number of intelligently generated points that allows LINDOGlobal to find a number of locally optimal points and report the best one found [18]. This feature allows the solver to be applied to multimodal functions. Once a optimization problem is submitted to NEOS, all additional information required by the optimization solver is determined automatically.

The LINDOGlobal output file does not provide any information about the number of function evaluations, then we only compared the optimal values given by the solver and those ones generated by our approach. For the  $q$ -gradient method we considered the parameters  $\sigma^0$ ,  $\alpha^0$  and  $\beta$  defined in Table 1. The stopping criteria are now the value of local gradient  $\|\nabla F(\mathbf{x}^k)\| < 10^{-20}$ , or the maximum of  $10^6$  function evaluations. For the LINDOGlobal, we implemented the test functions with the standard options and the same initial point set as in the  $q$ -gradient method. The best value for each method was taken among the 50 independent runs.

Results are shown in Table 4. For the unimodal functions, the deterministic solver, as expected, displays a better performance. However, for the multimodal functions, the  $q$ -gradient method attained a better accuracy and outperformed the LINDOGlobal solver.

**Table 4** The best solution found for GAMS/LINDOGlobal solver and  $q$ -gradient method over the test functions.

Functions	$q$ -gradient	LINDOGlobal
Ellipsoidal	0	0
Schwefel	4.4492E − 28	0
Rosenbrock	9.1939E − 11	0
Ackley	−4.4409E − 16	−6.0396E − 14
Rastrigin	0	−1.3358E − 12
Rotated Rastrigin	0	−1.2506E − 12

## 6 Conclusions and Future Work

In this work we introduced the  $q$ -gradient method for global optimization. The main idea behind the method is the use of the negative of the  $q$ -gradient of the objective function — a generalization of the classical gradient based on the Jackson’s derivative — as the search direction. The use of Jackson’s derivative provides us an effective mechanism for escaping from local minima. The algorithm is implemented in a way that the search process gradually shifts from global in the beginning to almost local search in the end.

For testing this new approach, we considered six commonly used 20-variable test functions. These functions display features of real-world optimization problems (multimodality, for example) and are notoriously difficult for optimization algorithms to handle. We compared the  $q$ -gradient method with GAs developed by Deb et al. [9], and Ballester and Carter [10] with promising results. Overall, the  $q$ -gradient method clearly beat the competition in the hardest test cases, those dealing with the multimodal functions.

It comes without surprise the (relatively) poor results of the  $q$ -gradient method with the Rosenbrock's function, a unimodal test function specially difficult to be solved by the steepest descent method. This result highlights the need for the development of a  $q$ -generalization of the well-known conjugate-gradient method, a research line currently being explored.

**Acknowledgements** The authors gratefully acknowledge the support provided by the National Counsel of Technological and Scientific Development (CNPq), Brazil.

## References

1. Ernst, T.: The history of  $q$ -calculus and a new-method. U.U.D.M. Report 2000:16, Department of Mathematics, Uppsala University, Sweden (2000)
2. Ernst, T.: A method for  $q$ -calculus. *J. Nonlinear Math. Phys.* 10, 487–525 (2003)
3. Chaundy, T.W.: Frank Hilton Jackson (obituary). *J. London Math. Soc.* 37, 126–128 (1962)
4. Jackson, F.H.: On  $q$ -functions and a certain difference operator. *Trans. Roy. Soc. Edinburgh* 46, 253–281 (1908)
5. Jackson, F.H.: On  $q$ -definite integrals. *Quart. J. Pure Appl. Math.* 41, 193–203 (1910)
6. Koekoev, J.: A note on the  $q$ -derivative operator. *J. Math. Anal. Appl.* 176, 627–634 (1993)
7. Kac, V., Cheung, P.: *Quantum Calculus*. Universitext, Springer New York, New York, NY (2002)
8. Di Pillo, G., Palagi, L.: Unconstrained Nonlinear Programming. In: Pardalos, J.M., Resende, M.G.C. (eds.) *Handbook of Applied Optimization*, pp. 268–284. Oxford University Press, New York (2002)
9. Deb, K., Anand, A., Joshi, D.: A computationally efficient evolutionary algorithm for real-parameter optimization. *Evolutionary Computation* 10-4, pp. 371–395 (2002)
10. Ballester, P.J., Carter, J.N.: An effective real-parameter genetic algorithm with parent centric normal crossover for multimodal optimisation. In: *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 901–913. Seattle, WA, USA (2004)
11. Czyzyk, J., Mesnier M., Mor J.: The NEOS Server. *IEEE Journal on Computational Science and Engineering*, 5, pp. 68–75, (1998)
12. Vanderplaats, G.N.: *Multidiscipline design optimization*. Vanderplaats Research and Development Inc., Monterey, CA (2007)
13. Nocedal, J., Wright, S.J.: *Numerical optimization* (2nd ed). Springer Science Business Media, New York, NY (2006)
14. Soterroni, A. C., Galski, R. L., Ramos, F. M.: The  $q$ -gradient vector for unconstrained continuous optimization problems. In: *Operations Research Proceedings 2010*, pp. 365–370. Springer-Verlag Berlin Heidelberg (2011)
15. Locatelli, M.: Simulated annealing algorithms for continuous global optimization. In: Pardalos, J.M. and Romeijn, H. E. (eds.) *Handbook of Global Optimization II*, pp. 204–207. Kluwer Academic Publishers, Dordrecht (2002)
16. Shang, Y., Qiu, Y.: A note on the extended Rosenbrock function. *Evolutionary Computation* 14, 119–126 (2006)
17. Rosenthal, R.E.: *GAMS - A User's Guide*. GAMS Development Corporation, Washington D.C. (2012)
18. LINDO Systems Inc.: *LINDOGlobal*. <http://www.gams.com/dd/docs/solvers/lindoglobal.pdf>